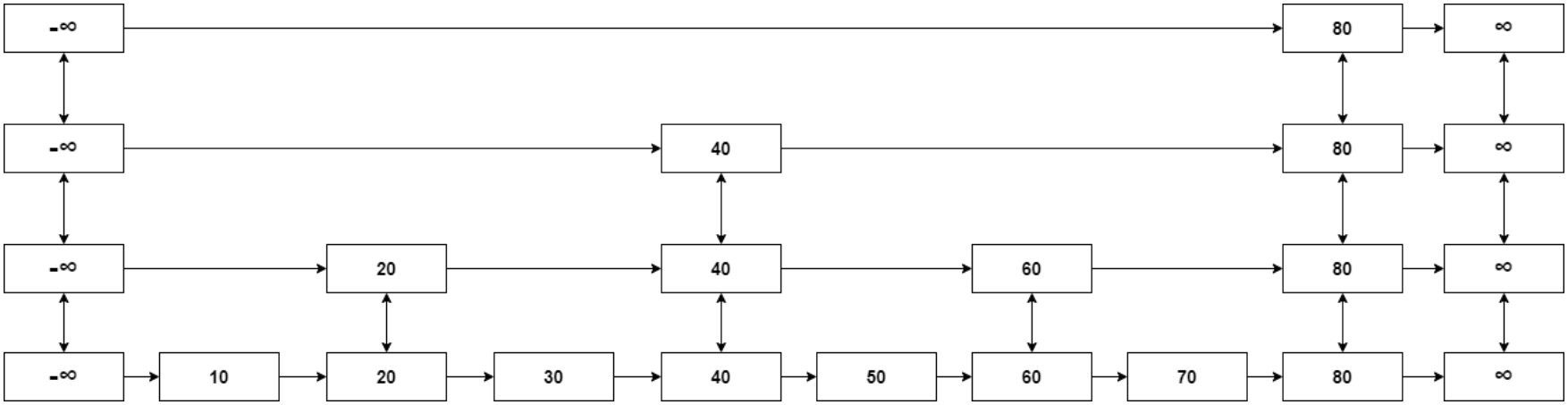
Group members: Marwane Bidamou (4), Eduard Koshkelyan (8), Roshan Maharjan (11)

Question 1. Skip List

Create a Skip list starting with no values and inserting each value one at a time. Result of the coin toss after inserting is also shown below. Based on this information, draw a skip list. How will you characterize the skip list produced?

Insert(10) T
Insert(20) HT
Insert(30) T
Insert(40) HHT
Insert(50) T
Insert(60) HT
Insert(70) T
Insert(80) HHHT

Ans:



Group members: Marwane Bidamou (4), Eduard Koshkelyan (8), Roshan Maharjan (11)

Question 2. Experimenting with lower bound

Devise an algorithm to sort 4 elements using exactly 5 comparisons in the worst case. Does this violate the theoretical lower bound? Justify your answer.

Ans:

Algorithm sortElements(A)
Input: An array A with 4 elements
Output: Sorted Array

```
if A[0] > A[1] then
        temp ← A[1]
        A[1] ← A[0]
        A[0] ← temp

if A[3] < A[2] then
        temp ← A[3]
        A[3] ← A[2]
        A[2] ← temp

if A[0] > A[2] then
        temp ← A[2]
        A[2] ← A[0]
        A[0] ← temp

if A[1] > A[3] then
        temp ← A[3]
        A[3] ← A[1]
        A[1] ← temp
```

Group members: Marwane Bidamou (4), Eduard Koshkelyan (8), Roshan Maharjan (11)

```
if A[1] > A[2] then
        temp ← A[2]
        A[2] ← A[1]
        A[2] ← temp

return A
```

This does not violate the theoretical lower bound because it needs at least n+1 to sort elements. Since it has to look at all elements to put it in order it has a time complexity of O(n).

Question 3

Definition
An array is said to be a FBS array if it satisfies the following three conditions.
(1) Elements in the odd locations are sorted in the ascending order.
(2) Elements in the even locations are sorted in the descending order.
(3) Every element in the even locations are <= every element in the odd locations.

Example {7, 20, 10, 19, 10, 17, 14, 15, 15}

Devise an algorithm to FBSsort (that will make an array a FBS array). What is the asymptotic running time of your algorithm? What is the fastest possible asymptotic running time for such an algorithm? Justify your answer.

Ans:

Algorithm FBSSort(A)
Input: Array of numbers
Output: Sorted FBS array

Group members: Marwane Bidamou (4), Eduard Koshkelyan (8), Roshan Maharjan (11)

1. Initialize two lists: `oddList = []`, `evenList = []`

2. Separate odd and even indexed elements:
   For each element A[i] in A:
   If i is odd, append A[i] to `oddList`
   Else, append A[i] to `evenList`

3. Sort `oddList` in ascending order.

4. Sort `evenList` in descending order.

5. Reassign sorted values back to their respective positions in array A:
   For each i in A:
   If i is odd, place the next element of `oddList` at A[i]
   If i is even, place the next element of `evenList` at A[i]

6. Check if the smallest element in `oddList` >= largest element in `evenList`:
   If this condition is violated, swap the largest even element with the smallest odd element.

7. Return A (the sorted FBS array)

The asymptotic running time is O(n log n).

Time Complexity

1. **Extracting odd and even elements**:
   - This requires scanning through the array once, which is **O(n)**.
2. **Sorting odd and even lists**:
   - Sorting the odd and even indexed elements takes O(n/2 log n/2), which simplifies to **O(n log n)**.
3. **Reassigning sorted values to the original array**:
   - This requires another pass through the array, which takes **O(n)**.

Group members: Marwane Bidamou (4), Eduard Koshkelyan (8), Roshan Maharjan (11)

4. **Checking and enforcing the third condition**:
    - This step involves scanning the odd and even lists and performing at most one swap, which takes **O(1)**.